

AMENDMENTS TO THE CLAIMS

This listing of claims replaces all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented method for arranging vector graphics data for processing into an output, the method comprising:
 - receiving a function call via an application programming interface (API) of a media integration layer (MIL),
 - the API providing access to the functionality of the MIL,
 - ~~the media integration layer being among a plurality of layers in a graphics processing environment,~~ the MIL media integration layer comprising a plurality of types of objects, the objects including a plurality of VisualManager objects, each VisualManager object which manages the rendering connecting a Visual Tree to a particular medium, each VisualManager object having a relationship with a window in which graphic data is output, and each VisualManager managing the rendering process to the particular medium,
 - ~~the function call comprising graphics-related data;~~
 - ~~a parser/translator interpreting the function call;~~
 - ~~comprising graphic-related data~~
 - connecting, by a VisualManager, a Visual Tree to a render target which is a particular medium, the Visual Tree comprising a plurality of Visuals, each Visual providing parent visual access, child visual collection, clipping, opacity, blendmode, transform, hit testing, and bounding box services;
 - ~~causing data in a scene graph data structure to be modified, the parser/translator enabled to interpret markup language data, in native format, direct code calls, object model code calls, and XML-based markup; and~~
 - traversing the Visual Tree;
 - rendering, by a VisualRenderer, the Visual Tree to the particular medium; and
 - causing a change in a graphics display of the particular medium in response to the modification of data in the scene graph.

2. (Previously Presented) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to initialize a new instance of a non-drawing visual class.

3. (Original) The method of claim 2 further comprising, receiving a function call via an interface corresponding to a transform associated with the visual.

4. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to initialize a new instance of a drawing visual class.

5. (Original) The method of claim 4 further comprising, receiving a function call via an interface to open the drawing visual instance for rendering, and in response, causing a drawing context to be returned, the drawing context providing a mechanism for rendering into the drawing visual.

6. (Original) The method of claim 1 further comprising, receiving brush data in association with the function call, and wherein causing data in a scene graph data structure to be modified comprises invoking a brush function to modify a data structure in the scene graph data structure such that when a frame is rendered from the scene graph, an area will be filled with visible data corresponding to the brush data.

7. (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a solid color.

8. (Previously Presented) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a linear gradient brush and a stop collection comprising at least one stop.

9. (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to a radial gradient brush.

10. (Original) The method of claim 6 wherein receiving brush data comprises receiving data corresponding to an image.

11. (Original) The method of claim 10 further comprising, receiving a function call via an interface corresponding to an image effect to apply to the image.

12. (Original) The method of claim 1 further comprising, receiving pen data in association with the function call, and wherein causing data in a scene graph data structure to be modified comprises invoking a pen function that defines an outline of a shape.

13. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent an ellipse in the scene graph data structure.

14. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent a rectangle in the scene graph data structure.

15. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent a path in the scene graph data structure.

16. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a geometry-related function to represent a line in the scene graph data structure.

17. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function related to hit-testing a visual in the scene graph data structure.

18. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function related to transforming coordinates of a visual in the scene graph data structure.

19. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function related to calculating a bounding box of a visual in the scene graph data structure.

20. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function via a common interface to a visual object in the scene graph data structure.

21. (Original) The method of claim 1 further comprising invoking a visual manager to render a tree of at least one visual object to a rendering target.

22. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place a container object in the scene graph data structure, the container object configured to contain at least one visual object.

23. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place image data into the scene graph data structure.

24. (Original) The method of claim 23 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place an image effect object into the scene graph data structure that is associated with the image data.

25. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place data corresponding to text into the scene graph data structure.

26. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to provide a drawing context in response to the function call.

27. (Original) The method of claim 26 wherein the function call corresponds to a retained visual, and further comprising, calling back to have the drawing context of the retained visual returned to the scene graph data structure.

28. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place a three-dimensional visual into the scene graph data structure.

29. (Original) The method of claim 28 wherein causing data in a scene graph data structure to be modified comprises mapping a two-dimensional surface onto the three dimensional visual.

30. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place animation data into the scene graph data structure.

31. (Original) The method of claim 30 further comprising communicating timeline information corresponding to the animation data to a composition engine at another layer of the media integration layer.

32. (Original) The method of claim 31 wherein the composition engine interpolates graphics data based on the timeline to animate an output corresponding to an object in the scene graph data structure.

33. (Original) The method of claim 1 wherein receiving a function call via an interface of a media integration layer comprises receiving markup, and wherein causing data in a scene graph data structure to be modified comprises parsing the markup into a call to an interface of an object.

34. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises invoking a function to place an object corresponding to audio and/or video data into the scene graph data structure.

35. (Original) The method of claim 1 wherein causing data in a scene graph data structure to be modified comprises changing a mutable value of an object in the scene graph data structure.

36 – 67: (Withdrawn)

68. (Previously Presented) A computer program product comprising one or more computer-readable media having computer-executable instructions encoded thereon which, when executed by suitable computing equipment, implement the method of claim 1.